# スツルム・同時逆反復法 (数値計算のアルゴリズムの研究)

AUTHOR(S):

村田, 健郎; 後, 保範

CITATION:

村田, 健郎 ...[et al]. スツルム・同時逆反復法 (数値計算のアルゴリズムの研究). 数理解析研究所講究録 1982, 453: 60-88

ISSUE DATE:

1982-02

URL:

http://hdl.handle.net/2433/102992

数理解析研究所講究録
453巻 1982年 60-88



# スツルム・同時逆反復法

村田 健郎（日立中研）
後 保範（日立ソフト）

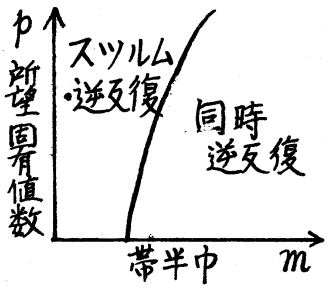
## 1 はじめに

対称帯固有値問題 $Ax=\lambda Mx$ を、帯のワクの中で解くことのできるアルゴリズムは、$Ax=\lambda x$ 形の問題とくらべると限られて来る．主なものは、

(1) スツルム・逆反復法 系統のもの （レーリー商シフト逆反復併用）

(2) 同時逆反復法 系統のもの （これにもスツルムチェックは必要）

である．他にランチョス法系統のものがあるが、これは魅力はあるが汎用化のむづかしさは抜群ゆえ、これを保留すると(1)，(2)が残る。



〈図 1〉

CPUタイムの観点からの両者の適用範囲を概念的に示すと、およそ図1のようになる。同時に、ページスワップについての負担の評価を行なわないと実用上片手落ちとなるが，それはこみ入ったことになり、今まで実用的な評価がなされたことがない。

(1)の系統についてのコメント ： 二分法の最小区間巾を決める eps値（本文の epsbi）と、シフト点固定の逆反復をどこまでやらせるか を適当に制御することによって、CPU

タイム，USEタイムの何れに重点を置くかの選択は可能であるが、密集固有値があるとき、何れか一方はどうしても長くなる．(1)

(2)の系統についてのコメント：　同時逆反復を何回かに分けて行なう提案がいくつかなされている．(2,3,4)　ところが密集固有値群が複数組あるような問題に対して、所望の固有値，固有ベクトルを、洩れなく精度よく求めようという観点からの検討や、ページスワップに対する配慮に欠けるところがあるため、汎用的なプログラム・プロダクトとしては俄かには採用し難い．

さて、今回のものは、前回報告のINVRAY(1)の弱点を改修するために自然に思いつかれたものであるが、要するに密集固有値をスツルムを使用して合理的に区分けして、そこに対しては同時逆反復法が働らくよう仕組まれたものである．即ち、(1)，(2)の長所を生かし、短所を表面化させないよう両者を統合したものとなった．その結果，

(イ)　帯巾が広く、かつ密集固有値があるため(1)では難儀な問題，

(ロ)　所望の固有値数が多く、かつ分布が精疎交々、しかも所望精度が精しいため(2)では難儀な問題

の双方に対し、CPU，USEタイム共にそれなりに耐えるようになった．

便宜上，プログラムをBISECT部とGINVRY部に分けた。

BISECT部では、いわゆるスツルム，即ち $A-\alpha M$ の負の固有値の個数‘count’を数えるためのプログラム：SSTRMを使用して、与えられた $al$ に対し次のことを行なう：

i) $0<\lambda_i<al$ なる $\lambda_i$ の個数 $nc$ を定める.

ii) $epsbi=(al/nc)*l$ （$l$ は2及至5，標準値は3）

iii) 固有値を、孤立，あるいはepsbi値以下の区間内に追いこむかの何れかになるまで分離.（粗いグループ分け）

GINVRY部 始めのところで、再度SSTRMを使ってグループ分けを再編成する．その後、各グループ毎に、alfa点を固定した単独，あるいは同時逆反復を行なう．反復回数の上限に達しても收束しないものは、最後にレーリ商シフトつき逆反復によって止めを刺す.

## 2 SSTRM

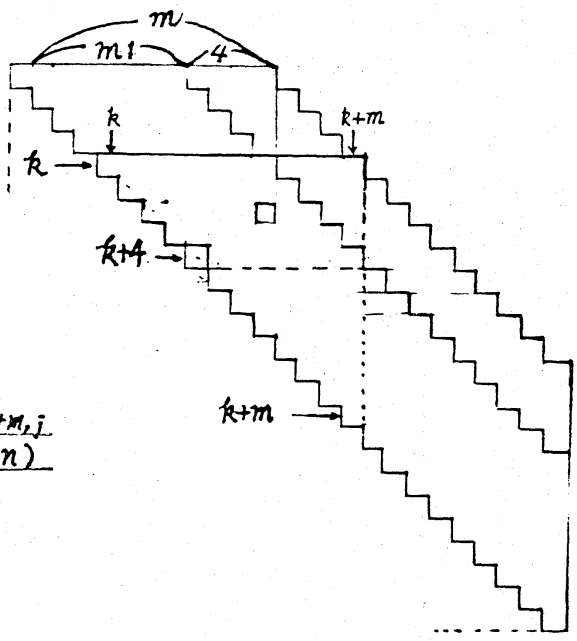
$A-\alpha M$の負の固有値の数を数へるアルゴリズムとしては、Martin-Wilkinsonの特殊ガウス[5],[1]がオーソドックスであるが、それによるCPUタイムの負担は大きい。ところで、我々の場合のepsbi（分割の精しさ）は、普通とくらべて甚だしく粗いから，‘数個先*までの対角要素の絶対値最大のものをピボットとする‘行・列交換の対称ガウス’で間に合う.

---

* ‘4個先’程度で実用上十分．このときSSTRM一回当りのCPUタイムは、約 $\frac{1}{2}(m1+4)^2n$ 積和，(Martin·Wilkinsonだと $2(m1+1)^2n$ 積和.)

# SSTRM {αを与えてA-αMの負固有値数nctを得る}

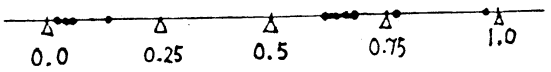
```
      m = m1+4 ; ir = 0 ; epsir = 0.2*10^-15
500   continue
      do i = 1, m
         do j = i, min(i+m1, n)
            aa[i,j] = a_ij - α*m_ij
         do j = i+m1+1, min(i+m, n)
            aa[i,j] = 0.0
      nct = 0   ; anorm = 0.0
      do k = 1, n
         do j = k+m, min(k+m+m1, n)
            aa[k+m,j] = a_k+m,j - α*m_k+m,j
         do j = k+m+m1+1, min(k+m+m, n)
            aa[k+m, j] = 0.0

         ipk = k ; amax = abs(aa[k,k])
         do i = k+1, min(k+4, n)
            aik = abs(aa[i,i])
            if aik > amax then
*              amax = aik ; ipk = i
         if ipk ≠ k then
            aa[k,k] ⇄ aa[ipk, ipk]
            do j = k+1, ipk-1
               aa[k,j] ⇄ aa[j, ipk]
            do j = ipk+1, min(k+m, n)
               aa[k,j] ⇄ aa[ipk, j]
         anorm = max(anorm, Σ_{j=k}^{jmax} |aa[k,j]| )    {jmax = min(k+m, n)}
         if amax < epsir * anorm then
            ir = ir+1 ; α = α + epsir*100^ir ; go to 500
         do j = k+1, min(k+m, n)
            ak[j] = aa[k,j]
         do i = k+1, min(k+m, n)
            t = ak[i] / aa[k,k]
            do j = i, min(k+m, n)
               aa[i,j] = aa[i,j] - t * ak[j]
         if aa[k,k] < 0.0  nct = nct+1
```




実際には、k行からk+m行までのaa[i,j]を、(1+m)×mの固定のアレイar[1+m,m]に收め、それをサイクリックに使って上の仕事を行なう。(文献(1)で述べた技法と同じ)

5

## 3 BISECT




```
      call SSTRM (alを与えてA-al*Mの負固有値数 ncを得る.)
      do i=1,nc
         xl[i] = al ; xu[i]= 0.0 ; nc[i] = nc
      epsbi = (al/nc ) * 3          {今回はepsbiを外から与える}
      k = 1
50    continue {repeatのstart point}
         if xl[k]-xu[k]≤epsbi go to 200
60       continue {repeatのstart point}
            alfa = (xu[k]+xl[k])/2.0
            call SSTRM {alfaを与えてnctを得る.}
            do i=k,nct
               xl[i] = alfa, nc[i] = nct
            do i=nct+1, nc
               xu[i] = max(alfa,xu[i])
            if (k<nc[k]) and((xl[k]-xu[k])>epsbi) go to 60
200      continue
         k = nc[k]+1
         if k≤nc go to 50
      do i=1,nc                     {これはデバッグ時のoption}
         write  i, nc[i], xu[i], xl[i]
```

出力例：これは、あとで示す問題A1の、al=1.0, epsbi=0.4 でのもの.

| i | nc[i] | xu[i] | xl[i] |
|---|---|---|---|
| 1 | 6 | 0.0 | 0.25 |
| 2 | 6 | 0.0 | 0.25 |
| 3 | 6 | 0.0 | 0.25 |
| 4 | 6 | 0.0 | 0.25 |
| 5 | 6 | 0.0 | 0.25 |
| 6 | 6 | 0.0 | 0.25 |
| 7 | 16 | 0.50 | 0.75 |
| 8 | 16 | 0.50 | 0.75 |
| 9 | 16 | 0.50 | 0.75 |
| 10 | 16 | 0.50 | 0.75 |
| 11 | 16 | 0.50 | 0.75 |
| 12 | 16 | 0.50 | 0.75 |
| 13 | 16 | 0.50 | 0.75 |
| 14 | 16 | 0.50 | 0.75 |
| 15 | 16 | 0.50 | 0.75 |
| 16 | 16 | 0.50 | 0.75 |
| 17 | 18 | 0.75 | 1.0 |
| 18 | 18 | 0.75 | 1.0 |

区間 (0, 0.25) に 6ケ, 区間 (0.50, 0.75) に 10ケ, 区間 (0.75, 1.0) に 2ケ の固有値があることが示されている.

# 4 GINVRY

初めに大筋を示す．簡単のため $M=I$ とする．

① 合理的なグループ分けを行う、SSTRM を用び使用
グループ番号 ig，固有値個数 ln[ig]，
グループ下限 xug[ig]，上限 xlg[ig]，igmax を決定

```
k=1
do ig = 1, igmax
```

② `alfa = (6.0*xug[ig]+ 4.0*xlg[ig])/10.0`
`kp1 = k-1+ ln[ig] ; lo = 1 ; ir1 = 0`

100 `continue` {やり直しのスタートポイント}

$v_k, \cdots, v_{kp1}$ に初期ベクトルを用意，但しお互同志のみならず、ig-1 グループの既求の固有ベクトルとも直交化

③ `lp = lo * max(2, ln[ig]/2 +1)` {反復回数の単位}
`lq = 0`

199 `continue`
`lq = lq+1`

④ lp 回 同時(または単純)逆反復，$\bar{v}_i$ ($i=k,\cdots,kp1$) を求める

```
do i = k, kp1
```

$\bar{v}_i$ によるレーリー商により $\bar{\lambda}_i$ を求める．
`difoi = ‖A v̄i − λ̄i v̄i‖ / ‖A‖`

⑤ `if (lq<lqm)∧(difoi>epso)∧(k≠kp1) go to 199`

```
do i = k, kp1
```

`difi = difio ; ll = 1`

⑥ `while difi > eps ∧ ll ≤ llmax do`

⑦ レーリー商シフトつき逆反復で $v_i$ と $\lambda_i$ を求める

`difi = ‖A vi − λi vi‖ / ‖A‖`
`ll = ll+1`

⑧ {check ルーチン}. $xug[ig] \leqslant \lambda_i \leqslant xlg[ig]$ ($\forall i=k,\cdots,kp1$) なら OK
そうでなく、かつ ir1=0 なら `ir1=ir1+1, lo=2` として go to 100

`k = k+ln[ig]` {k を次グループの先頭に進める}

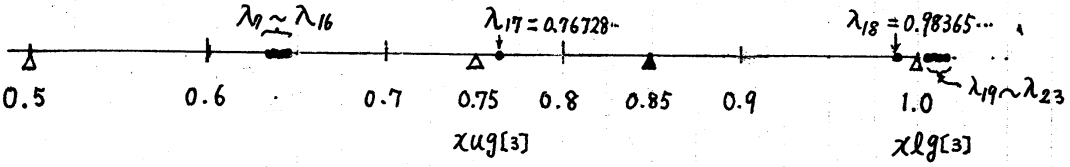
〔例〕 $epso = 0.2*10^{-15}$, $eps = 10^{-15}$, $lqm=10$, (epso < eps に注意)
$10^{-14}$, $10^{-13}$, $lqm=8$ ( 全上 )

① の部分

```
      k=1 ; nls=0 ; ig=0 ; h1=min(epsbi*0.1,(al/nc)*0.3)
      k=nc[k] ; lnig=nc[k]-nls ; igadv=1
      Call SSTRM {h1を与えてcountを得る}
      if count=0 then
          xu(k)=h1
      else
          if count=nc[k] xl[k]=h1
      ig=ig+1 ; ln[ig]=lnig ; xug[ig]=xu[k] ; xlg[ig]=xl[k]
      nls=nc[k]
      if epsbi≥al go to 12
      k=k+1
10    continue    {repeatの先頭}
          k=nc[k] ; lnig=nc[k]-nls ; igadv=1
          if xu[k]=xlg[ig] then
              alfa=xlg[ig]-h1
              Call SSTRM {alfaを与えてcountを得る}
              if count=nls then
                  xlg[ig]=alfa
              else
                  if count=nls-ln[ig] xug[ig]=alfa
                  alfa=xlg[ig]+h1 ;
                  Call SSTRM {alfaを与えてcountを得る}
                  if count=nls then
                      xu[k]=alfa
                  else igadv=0 ; if count=nc[k] xl[k]=alfa
          if igadv=1 then
              ig=ig+1 ; ln[ig]=lnig ; xug[ig]=xu[k] ; xlg[ig]=xl[k]
          else
              ln[ig]=ln[ig]+lnig ; xlg[ig]=xl[k]
          nls=nc[k] ; k=k+1
          if k≤nc go to 10
12    continue
      alfa=al-h1
      Call SSTRM {alfaを与えてcountを得る}
      if count=nc then
          xlg[ig]=alfa
      else
          alfa=al+h1
          Call SSTRM {alfaを与えてCountを得る}
          ln[ig]=ln[ig]+count-nc
          if count≠P xlg[ig]=alfa
      igmax=ig
```

(1)の部分がGINVRYの核心部である.

—BISECTの結果だけによってグループ分けしたのでは、例えばal=2.0, epsbi=0.4のとき、下図のようにグループ分けされるが、牙3グループがまずい.




それが、(1)の部分を通るとxlg[3]=1.2となってうまく行く.

(2) シフト矣alfaは、やゝ中矣より左寄りにとっている.

(3) 例えばln[ig]=10のとき、lp=1*max(2,6)=6となる.

(4),(5) (シフト矣から一番遠い)収束の遅い固有値だけは、収束するのを待たないで同時逆反復を打ちきる。lqmの大体の目安は、eps=10^-qのとき、lqm=max(6, q-6).

(6) すでにdifi<epsとなったものは(7)をやらせない。ここに、epsoは、epso<epsにとっておく。(例えばepso=10^-13, eps=10^-12). epso=epsにとると、次の(7)で、停滞することが稀にあるからである.

(7) ここの逆反復をコレスキーでやらせると、非常に稀なケースながら、失敗することがある.(epsが精しく、epsbiが適値より過大, lqm過少、かつ同時逆反復のシフト矣に対し密集固有値が離れている, といった状況のとき.)

同時逆反復　Rutishauserのアルゴリズム* を、シフト点を$\alpha$としたAx=λMx 用に直すと、次のようになる.（左側）これは、$Ax=\lambda Mx$ を $M=V^TV$, $z=Vx$ より $V^{-T}AV^{-1}z=\lambda z$ とし、これをシフト点を$\alpha$として下の右のアルゴリズムを作り、それをx系に戻したものである.

$$\begin{cases} (A-\alpha M)\bar{X}=MX^{\nu-1} \\ H=\bar{X}^TM\bar{X} \\ H=Q\Lambda Q^T \\ X^\nu=\bar{X}Q \\ \nu=\nu+1 \end{cases} \qquad \begin{cases} (V^{-T}AV^{-1}-\alpha I)\bar{z}=z^{\nu-1} & (1) \\ H=\bar{z}^T\bar{z} & (2) \\ H=Q\Lambda Q^T & (3) \\ z^\nu=\bar{z}Q & (4) \\ \nu=\nu+1 & (5) \end{cases}$$

（例えば(1)右より $(AV^{-1}-\alpha V^T)\bar{z}=V^Tz$, ここで $z=Vx$ とすれば(1)左を得る. 他も同様である.）

実際には、演算回数を減らすために次のようにする.

$$\begin{cases} (A-\alpha M)\bar{X}=Y^{\nu-1} & (6) \\ Y=M\bar{X} & (7) \\ H=\bar{X}^TY & (8) \\ H=Q\Lambda Q & (9) \\ Y^\nu=YQ & (10) \\ \nu=\nu+1 & (11) \end{cases}$$

(6)は実際には $A-\alpha M=LU$ を使って

$$LX=Y^{\nu-1}$$

$$U\bar{X}=X$$

とするが、このとき L, U のメモリエリアの方が、$Y^{\nu-1}$, X, $\bar{X}$ のそれと比べ断然大きいから、すべてのベクトルに対する求解を1回のL, U参照で済ますようにプログラムする.

* Parlettの本(6)の14章のIMPLEMENTATION NO.3

さて、シフト点$\alpha$が、たまたまある固有値$\lambda_i$に極度に近接しているとき、同時逆反復の対象となったすべてのベクトルが、一斉に$u_i$の方向を向いてしまうことが起り得る．これをさけるために、LU分解の中途でピボット値がある値以下になったら$\alpha$を動かしてやり直さねばならぬ．（SSTRMのときと同様である．）

この手当は、オーソドックスな部分軸選択のガウスを使用したとしても必要である．実用上は、SSTRMのときと同じ、'数仰先まで調べる対角上軸選択の行・列交換の対称ガウス'で十分である．（今、故意に全く軸選択を行なわない対称ガウスでテストを続けているが、所望の精度epsoを$0.2*10^{-15}$，即ち計算機精度ぎりぎりに設定したとき、$epso=10^{-13}$のときと比べて、やゝ異常に同時逆反復回数が増すことが、ときたまある程度である．後で示す問題A1でだけ起っている．）

---

〔附記〕 Parlettの本の14章のIMPLEMENTATION NO.4に従うと(2)の代りに、$\bar{z}=\bar{Q}R$，$H=RR^T$とすることになる．この方法は、$Ax=\lambda x$形の問題に対しては素直に適用できるが、$Ax=\lambda Mx$形の問題に対しては、X系に戻す手順がむづかしいので敬遠した．（Wilkinson Reinschの HAND BOOKにのっているのは NO.4の方である．）

(7)で使うガウスとしては、SSTRMで採用したアルゴリズムと本質的には同じものを使う。但し今度は、

$ar[1+j-i, i] = a_{ij} - \alpha \cdot m_{ij}$　　（今回のテストでは $m_{ij}=\delta_{ij}$）

の形に收めた帯アレイ $ar[m,n]$ を起用する。＊印のところまでは原理的にはSSTRMと同じだからその後を示す。

GUSBR（$b=\bar{v}_i$, $\alpha=\bar{\lambda}_i$ を与えて $(A-\alpha M)v_i=\bar{v}_i$ なる $v_i$ を $b$ に立てる。）

```
*  ここまではSSTRMと同じ。但し、ar[1,k]=aa[k,k], ar[1+j-i,i]=aa[i,j] etc.

   ip[k] = ipk
   if ipk ≠ k then
      ar[1,k] ⇄ ar[1,ipk]
      do j = k+1, ipk-1
         ar[1+j-k,k] ⇄ ar[1+ipk-j,j]
      do j = ipk+1, min(k+m,n)
         ar[1+j-k,k] ⇄ ar[1+j-ipk,ipk]
      b[k] ⇄ b[ipk]
   if amax < epsir  ar[1,k] = SIGN(epsir, ar[1,k])

   do j = k+1, min(k+m,n)
      ak(j) = ar(1+j-k,k)
   do i = k+1, min(k+m,n)
      t = ak(i)/ar[1,k]
      b[i] = b[i] - t*b[k]
      do j = i, min(k+m,n)
         ar[1+j-i,i] = ar[1+j-i,i] - t*ak[j]

   do k = n,1,-1
      S = 0.0
      do j = k+1, min(k+m,n)
         S = S + ar[1+j-k,k]*b[j]
      b[k] = (b[k] - S)/ar[1,k]
      if ip[k] ≠ k then
         B[k] ⇄ B[ip[k]]
```

この後、レーリー商 $v_i^T A v_i / v_i^T M v_i$ は、$\alpha + v_i^T (A-\alpha M) v_i / v_i^T M v_i$ に等しいことと、$(A-\alpha M) v_i = \bar{v}_i$ に留意して、

$$\text{新}\ \lambda_i = \alpha + v_i^T \bar{v}_i / v_i^T M v_i$$

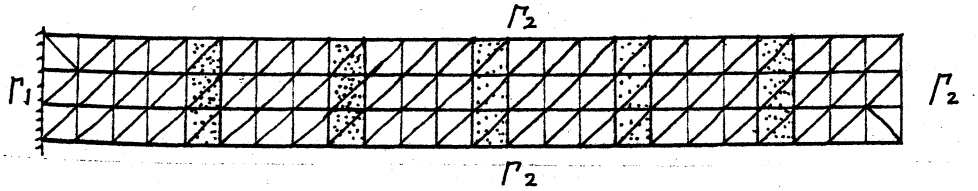
とする。尚、$w = M v_i$, 新 $\bar{v}_i = w / \|w\|$ である。新 $\lambda_i$ と $v_i$ による difi によって収束判定を行い、まだなら、$\alpha =$ 新 $\lambda_i$, $b =$ 新 $\bar{v}_i$ として用び GUSBR により逆反復を行なう。（反復が1回を超えることは殆んどない。これが2回以上のものが多数あるようでは、よほど初期ベクトルが不適当にえらばれているか、同時逆反復の lgm が過小かと疑うべきである。）

## 5 テスト例

固有値解析プログラムのデバッグと評価を行うに際して、決定的に重要なのは、テスト問題の選択である。いろいろのレベルの近接度をもった固有値系を、システマティックに作るために有限要素法を利用した。即ち、下図のような場を作り、▨部の k 値を任意に変えて、思いのまゝの近接度の固有値系をもった行列を生成した。（□部の k 値を 1.0 とする。）

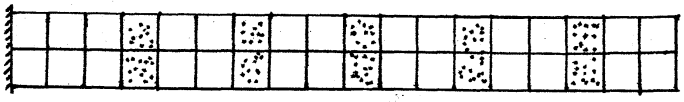
支配方程式は、$\nabla \cdot (-k(x) \nabla u) = \lambda u \Rightarrow A \underset{\sim}{u} = \lambda M \underset{\sim}{u}$ *

$\Gamma_1$ : 固定境界条件 $u = 0$ , $\Gamma_2$ : 自由境界条件 $\nabla u = 0$

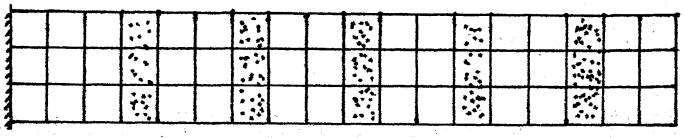



* 但し、今回は、$M = I$ としてテストを行なった。

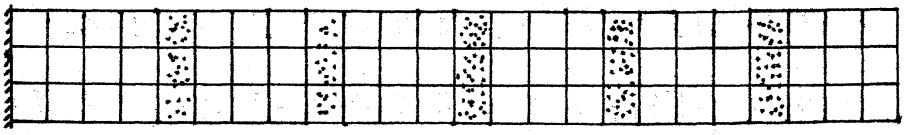
A



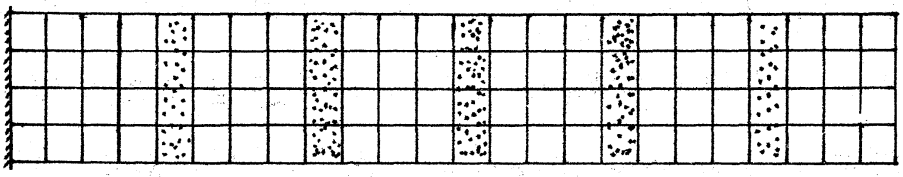
B



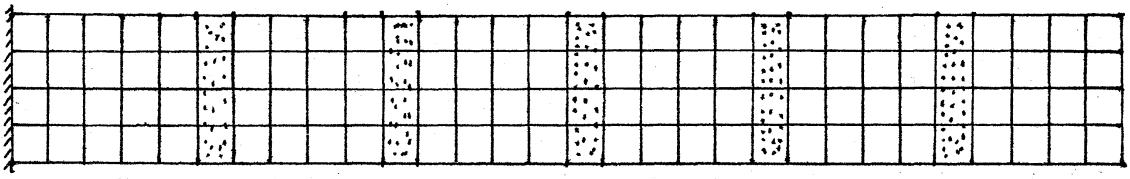
C



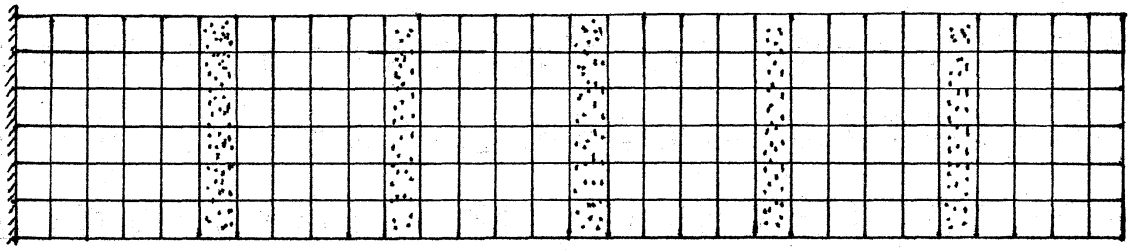
D



E



F



拡散係数 $k(i)$ 値をいろいろに与えて、固有値の密集度の異なる問題を作る.

上図で、□部の $k$ を 1.0 とし、▨部の $k$ を左から、

| | | | | | |
|---|---|---|---|---|---|
| 問題 1 | $10^{-2}$ , | $10^{-5}$ , | $10^{-5}$ , | $10^{-8}$ , | $10^{-8}$ |
| 問題 2 | $10^{-1}$ , | $10^{-3}$ , | $10^{-3}$ , | $10^{-5}$ , | $10^{-5}$ |
| 問題 3 | $10^{-1}$ , | $10^{-2}$ , | $10^{-2}$ , | $10^{-3}$ , | $10^{-3}$ |
| 問題 4 | 0.31 , | $10^{-1}$ , | $10^{-1}$ , | $10^{-2}$ , | $10^{-2}$ |
| 問題 5 | 0.5 , | 0.25 , | 0.25 , | 0.1 , | 0.1 |

例えば A の場で物質定数 $k$ を問題 1 のように えらんだとき、問題A1 と略称する.

問題A1の特徴 ⊞ の形の場が、5枚，弱いカップルでつながっていることから容易に想像されるように、5個一組の近接固有値の群ができるであろう．特にこの図形 ⊞ は、90°回転すると重なることから、自分自身だけなら重複固有値をもつ。それが、弱いカップルによって擾動を受け、二つの近接固有値にわかれるであろうから、5×2＝10個の近接固有値の群がときにあらわれるであろう．下に示す出力例は，$al = 1.5$ ($nc=23$), $epsbi = 0.4$, $epso = 0.2 * 10^{-15}$, $eps = 10^{-15}$, $lqm = 9$ によるもの．3グループに分けて解いている．

| i | ep | ln[ig] | xug[ig] | $\bar{\lambda}_i$ | alfa | $\lambda_i$ | difoi | xlg[ig] | difi |
|---|---|---|---|---|---|---|---|---|---|
| | 6 | 0.0 | | | 0.142173913 | | | 0.355434782 | |
| 1 | 4 | 0 | 0.134579834256475 | | 0.134579834256475 | | 0.27D-15 | | 0.27D-15 |
| 2 | 4 | 0 | 0.002146879526848 | | 0.002146879526848 | | 0.31D-15 | | 0.31D-15 |
| 3 | 4 | 0 | 0.000005816742572 | | 0.000005816742572 | | 0.28D-15 | | 0.28D-15 |
| 4 | 4 | 0 | 0.000000849770852 | | 0.000000849770852 | | 0.34D-15 | | 0.34D-15 |
| 5 | 4 | 0 | 0.000000005814621 | | 0.000000005814621 | | 0.41D-15 | | 0.41D-15 |
| 6 | 4 | 0 | 0.000000000847585 | | 0.000000000847585 | | 0.39D-15 | | 0.39D-15 |
| | 0 | | 0 | 0 | $l_q \to 8$ | | 32 | 192 | |
| | 10 | 0.375000000 | | | 0.517173913 | | | 0.730434782 | |
| 7 | 6 | 0 | 0.633974597550947 | | 0.633974597550947 | | 0.26D-14 | | 0.26D-14 |
| 8 | 6 | 0 | 0.633974599342072 | | 0.633974599342072 | | 0.13D-14 | | 0.13D-14 |
| 9 | 6 | 0 | 0.633974604384456 | | 0.633974604384456 | | 0.27D-14 | | 0.27D-14 |
| 10 | 6 | 0 | 0.633974604918049 | | 0.633974604918049 | | 0.46D-14 | | 0.46D-14 |
| 11 | 6 | 0 | 0.633975934590936 | | 0.633975934590936 | | 0.48D-15 | | 0.48D-15 |
| 12 | 6 | 0 | 0.633977726061515 | | 0.633977726061515 | | 0.26D-14 | | 0.26D-14 |
| 13 | 6 | 0 | 0.633982767790362 | | 0.633982767790362 | | 0.38D-14 | | 0.38D-14 |
| 14 | 6 | 0 | 0.633983299291280 | | 0.633983299291280 | | 0.20D-14 | | 0.20D-14 |
| 15 | 6 | 1 | 0.637156013823869 | | 0.637156013823869 | | 0.31D-13 | | 0.72D-15 |
| 16 | 6 | 1 | 0.637806121307539 | | 0.637806121307539 | | 0.32D-13 | | 0.78D-14 |
| | 6 | | 0 | 0 | 9 | | 54 | 540 | |
| | 7 | 0.750000000 | | | 1.042173913 | | | 1.480434782 | |
| 17 | 4 | 0 | 1.005093127020050 | | 1.005093127020050 | | 0.58D-16 | | 0.58D-16 |
| 18 | 4 | 0 | 1.000011035287053 | | 1.000011035287053 | | 0.18D-15 | | 0.18D-15 |
| 19 | 4 | 0 | 1.000003968597634 | | 1.000003968597634 | | 0.15D-15 | | 0.15D-15 |
| 20 | 4 | 0 | 1.000000011034313 | | 1.000000011034313 | | 0.11D-15 | | 0.11D-15 |
| 21 | 4 | 0 | 1.000000003964257 | | 1.000000003964257 | | 0.18D-15 | | 0.18D-15 |
| 22 | 4 | 0 | 0.983654899202529 | | 0.983654899202529 | | 0.15D-15 | | 0.15D-15 |
| 23 | 4 | 1 | 0.767280467691874 | | 0.767280467691749 | | 0.34D-06 | | 0.14D-15 |
| | 10 | | 0 | 0 | 4 | | 16 | 112 | |

BISECT のSSTRM 4　GINVRY SUBSPACE のSSTRM 4 3　LF（レーリーシフト反復） 3

問題C1，問題E1も同様である．

CPUカウント．

対称ガウス1回当りの $\frac{1}{2}m^2n$ 積和をCPUタイム概算のための1カウントとする．前頁の例で、SSTRMをBISECTのために4，GINVRYの初めのところで4，SUBSPACEのためのLU分解のため3，最後のレーリーシフト逆反復のため3，計14カウントまでは問題無い．シフト戻固定の同時（又は単独）逆反復の初めに行なう初期ベクトル作成のための直交化と正規化のために約 $(2\,ln[ig-1]+ln[ig])\,ln[ig]\cdot n + ln[ig]\,n$ 回積和を行なうが、我々にとって興味のある応用分野：$m \gg ln[ig]$ においては、これは $\frac{1}{2}m^2n$ 積和に対して無視してよい．

同時逆反復の内部は問題である．$p_i = ln[ig]$ として1反復当り積和回数は、$M = I$ のとき*

| $(A-\alpha I)\,\bar{X} = X^{\nu-1}$ | $H = \bar{X}^T\bar{X}$ | $= Q\Lambda Q^T$ | $X^\nu = \bar{X}Q$ |
|---|---|---|---|
| $2mnp_i$ | $np_i^2$ | $\sim p_i^3$ | $np_i$ |

$m > 50$，$p_i < 10$ という場合を問題にしているから、この表の第1項だけ考えればよい．* 逆反復を $\nu_i$ 回行なうとき、

$$2mnp_i\nu_i \;/\; \tfrac{1}{2}m^2n = 4p_i\nu_i/m \quad (\text{カウント})$$

を各ig毎に加えればよい．$m = 80$ のときこれは $p_i\nu_i/20$，$m = 160$ だと $p_i\nu_i/40$ となる．

* $M \neq I$ で、$H = \bar{X}^T M \bar{X}$ のときでも、通常Mは帯かっ帯の中も疎ゆえ大差ない．

## USE カウント

簡単のために、与えられる主実メモリ容量に比べて、$mn$ 語の方がはるかに大きいとする。現在の16メガバイトマシンで、マルチジョブ運転で一つのプログラムに与えられる主メモリ容量は、（深夜を除けば）高々2メガバイトである。$m=100$, $n=10{,}000$ の二次元問題ですでに $mn$ 語＝1メガ語すなわち8メガバイトである。

対称ガウスによる前進、後退の $2mn$ 語参照をUSEの1カウントとする。LU分解の方は $mn$ 語参照ゆえ $\frac{1}{2}$ カウントである。$A-\alpha M$ を作りながら分解を行なうとし、A, M自身はパックして收めていて、1行当りの非ゼロ要素の数 s は、$s \ll m$ ゆえ、A, M参照はカウントに入れぬことにする。

SSTRM のカウントは我々のプログラムによる場合、無視してよい。

同時逆反復については、$p_i = \ln[ig]$ として $\nu_i$ 回反復のとき、一つのグループ当り、$M=I$ のとき、

$(A-\alpha I)\bar{X} = X^{\nu-1}$ : $2(m+p_i)n\nu_i/2mn = \nu_i + (p_i\nu_i/m)$

$H = \bar{X}^T\bar{X}$ : $np_i\nu_i/2mn = (p_i\nu_i/2m)$

$X^\nu = \bar{X}Q$ : $np_i\nu_i/2mn = (p_i\nu_i/2m)$

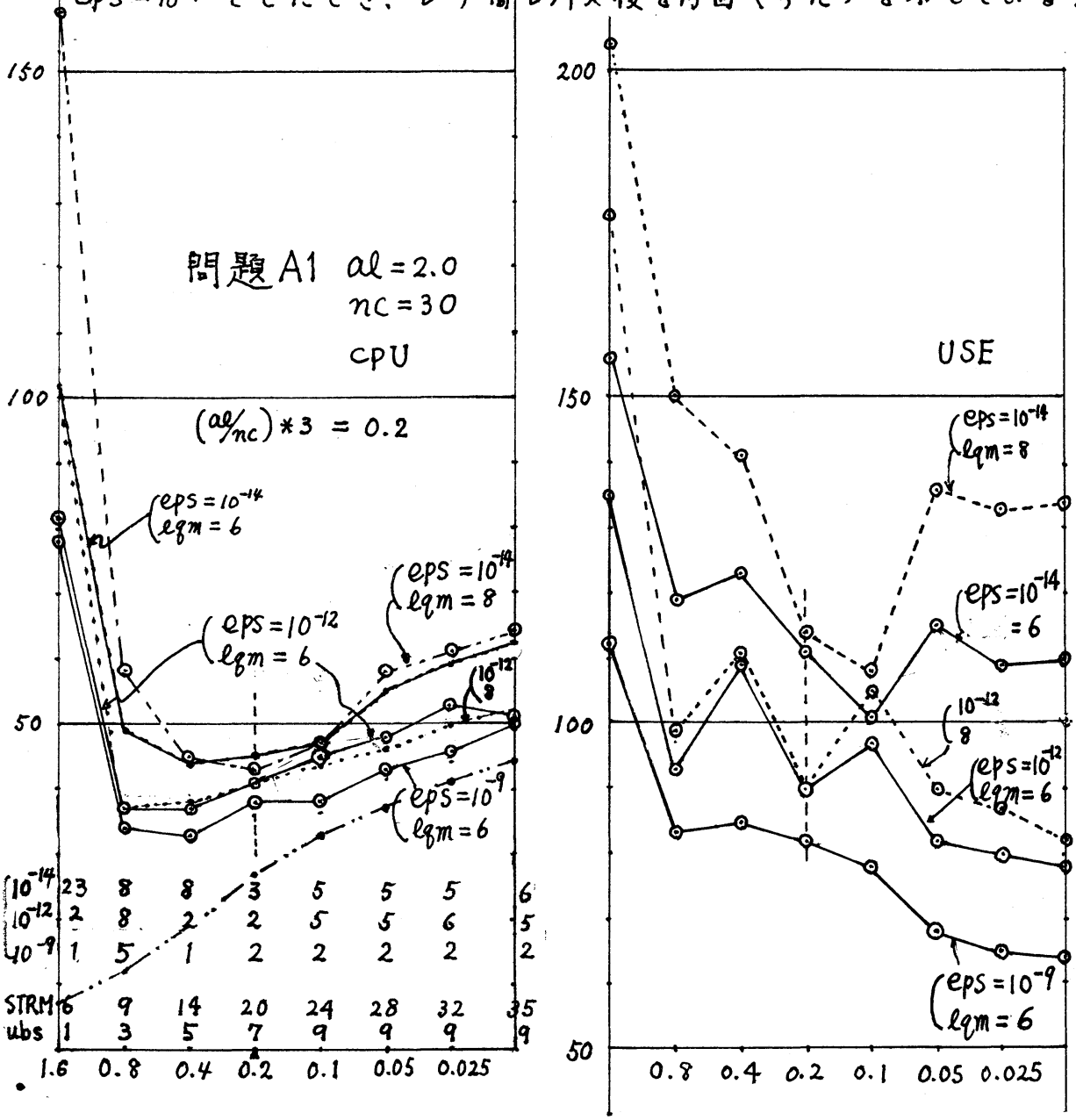
合計 : $\underbrace{\nu_i + (2p_i\nu_i/m)}$

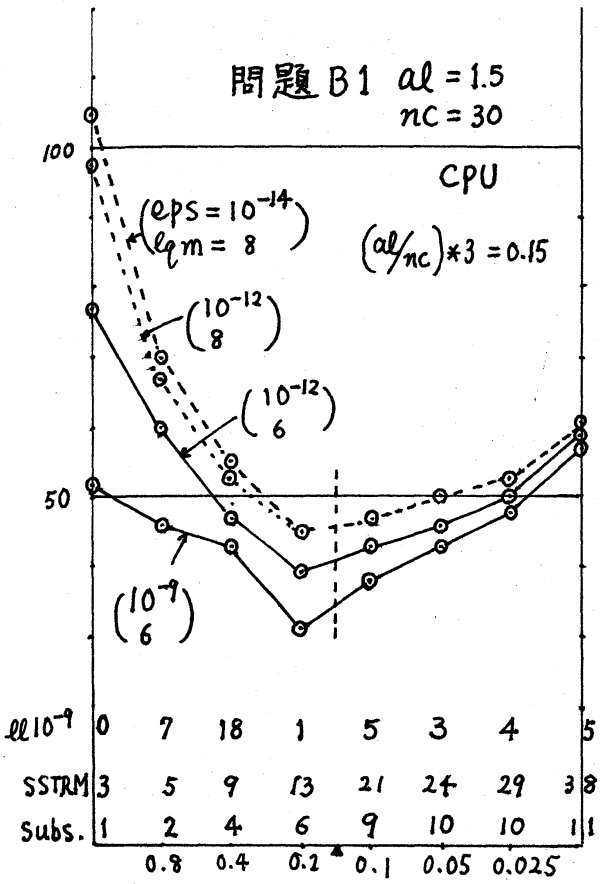
とする。（$M \neq I$ のときは、$H = \bar{X}^T\bar{Y}$ が $2np_i\nu_i = (p_i\nu_i/m)$ となる。）

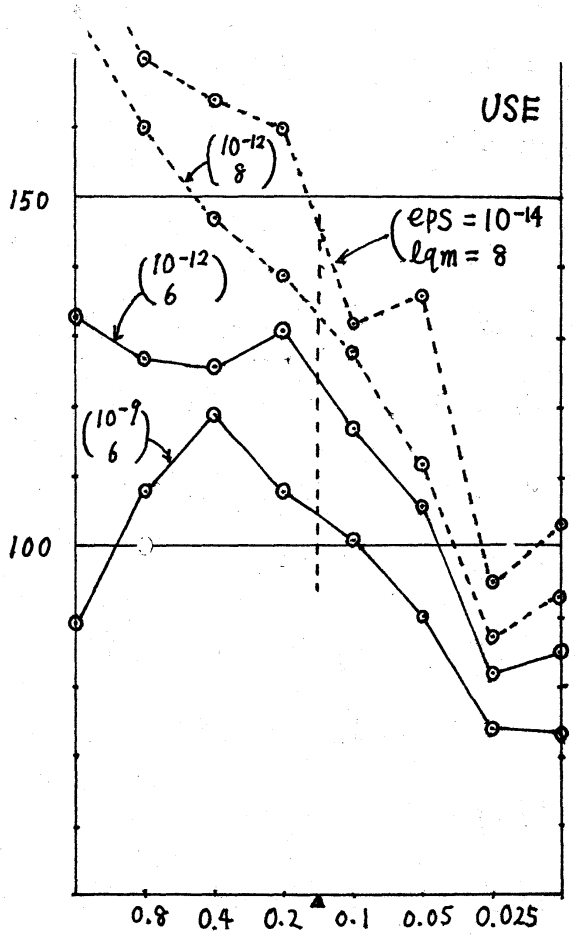
下の図1は 問題A1のデータにもとづいて、これと同じ固有値をもった $m=160$ の帯行列でCPUカウントとUSEカウントがどうなるであろうかを示すために作ったものである.

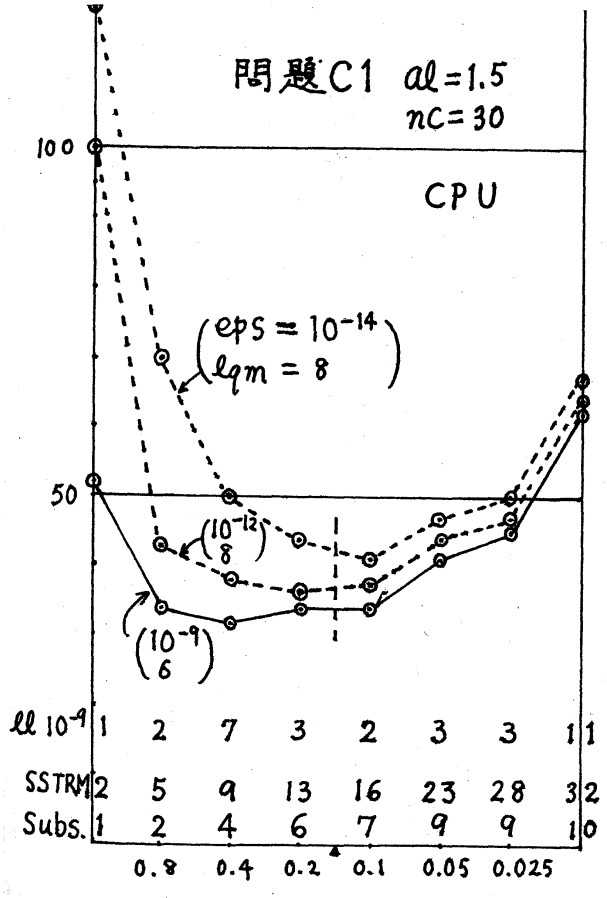
問題A1の固有値の分布は、やゝ極端である. 次頁に問題B1と問題C1によるものを示す.

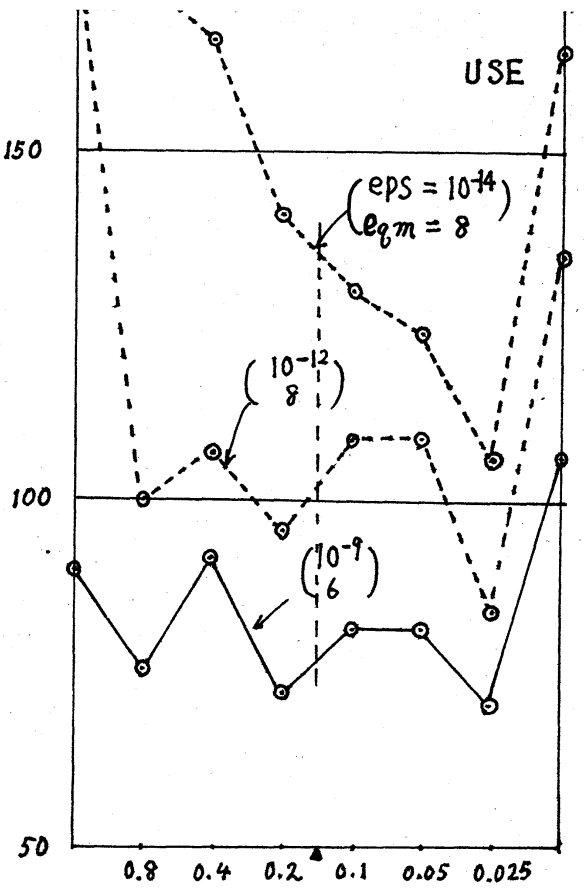
図の下段に Subs. とある行に書かれた数は、いくらのグループに分けたかを示す数字, SSTRM の行は SSTRM回数, ll の行は、$10^{-9}$ の行が, $eps=10^{-9}$ としたとき、レーリー商シフト反復を何回やったかを示している.




問題B1 al = 1.5
nc = 30
CPU
(al/nc)*3 = 0.15
(eps = 10^-14, lqm = 8)
(10^-12, 8)
(10^-12, 6)
(10^-9, 6)
100
50
ll 10^-9 0 7 18 1 5 3 4 5
SSTRM 3 5 9 13 21 24 29 38
Subs. 1 2 4 6 9 10 10 11
0.8 0.4 0.2 0.1 0.05 0.025


USE
(10^-12, 8)
(eps = 10^-14, lqm = 8)
(10^-12, 6)
(10^-9, 6)
150
100
0.8 0.4 0.2 0.1 0.05 0.025


問題C1 al = 1.5
nc = 30
CPU
(eps = 10^-14, lqm = 8)
(10^-12, 8)
(10^-9, 6)
100
50
ll 10^-9 1 2 7 3 2 3 3 11
SSTRM 2 5 9 13 16 23 28 32
Subs. 1 2 4 6 7 9 9 10
0.8 0.4 0.2 0.1 0.05 0.025


USE
(eps = 10^-14, lqm = 8)
(10^-12, 8)
(10^-9, 6)
150
100
50
0.8 0.4 0.2 0.1 0.05 0.025

次頁のものは、問題D1，D2，D3のデータに基いて作ったものである．最下段D3から作成したものには、D1，D2からのものも併せ記入してみた．$eps = 10^{-9}$，$lgm = 6$，$al = 1.2$でのもの．

D1のものは、たまたまグループ分けが特にうまく行って．同時（或いは単純）逆反復回数，レーリーレットっき逆反復共に特にすくなくて済んだときの例であり、D2，D3のものは逆にグループ分けがたまたまあまり好適でないときの例となっている．

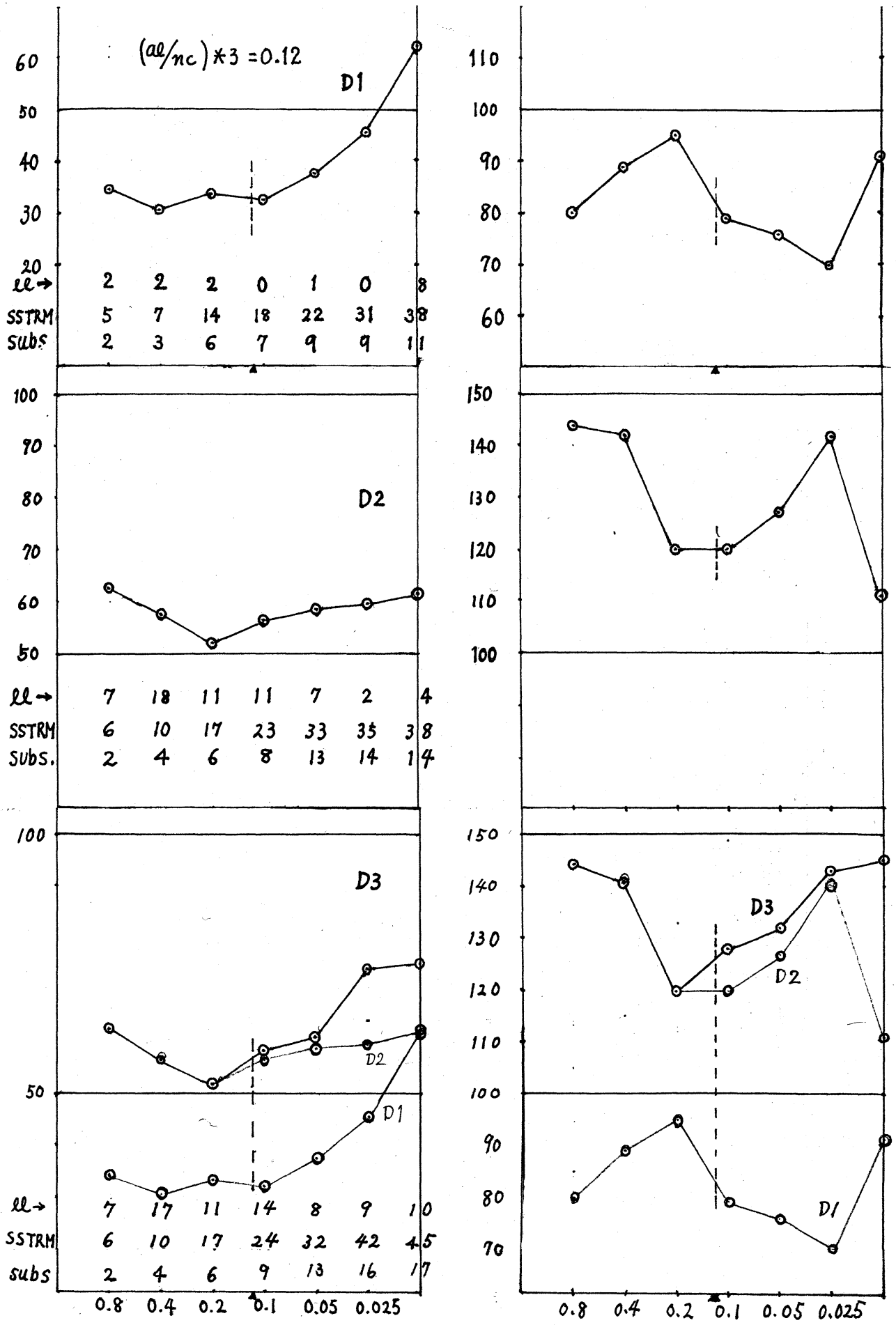
以上何れの場合にも、epsbiに関する設計上の最適値：

$$epsbi = (al/nc) * 3$$

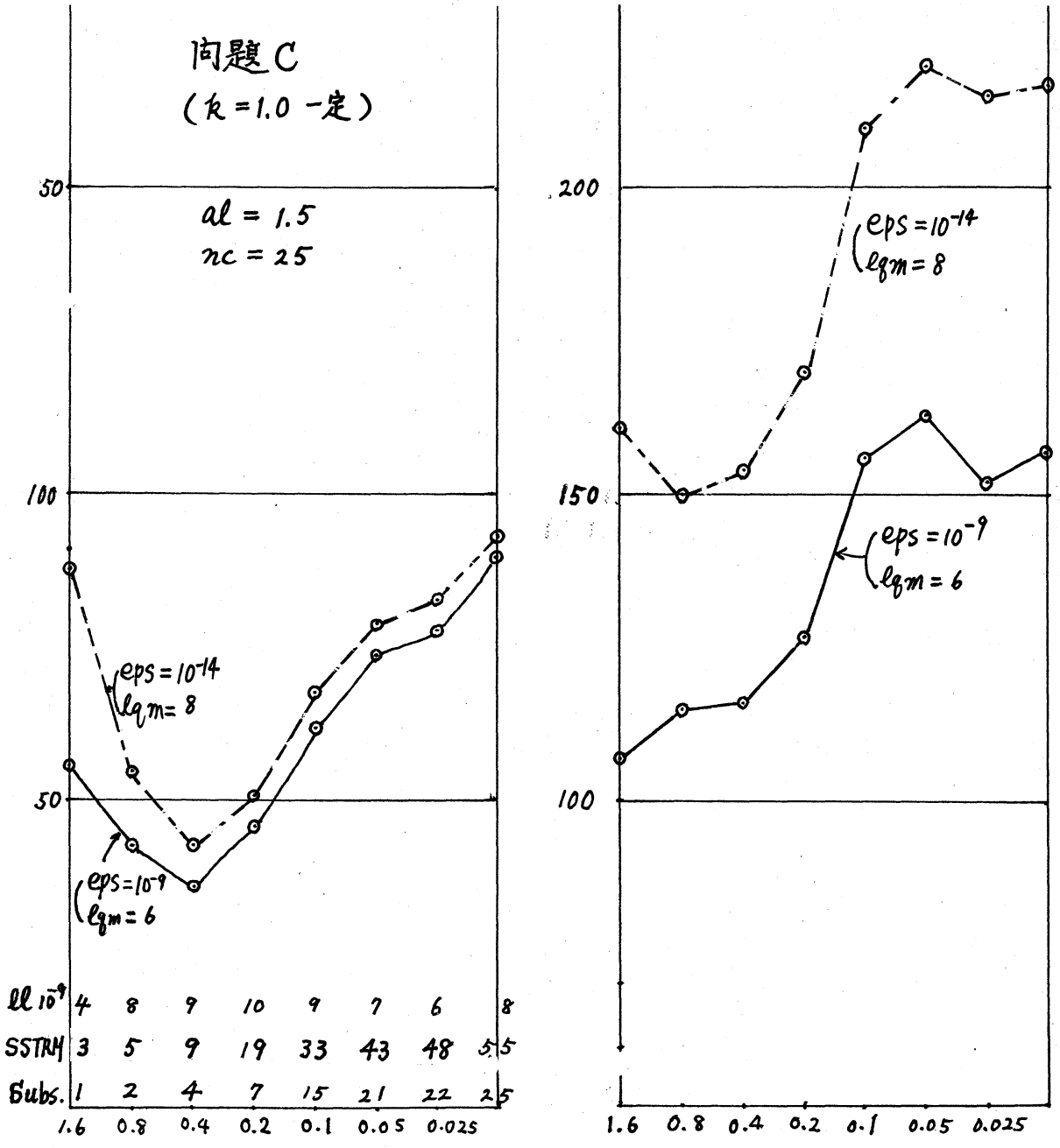
は、ほぼ妥当であることが示されている．但し、帯半巾mが40程度といった問題に対しては、epsbiの適値はもっと小さく，$epsbi = (al/nc) * 2$程度にすべきであろう．また、ここに示した例よりも固有値の分布が一様に近い（k(x)値の変化のすくない、例えば：問題4とか5の）問題に対しては、epsbiをや、大きくえらぶ．（例えば $epsbi = (al/nc) * 5$ 程度）

大体，一つのグループ当りの平均の固有値数が5前後になるようepsbiが えらばれたときがよいようである．問題1，2のような、固有値がそこここに密集しているような問題に対しては、$epsbi = (al/nc) * 3$程度にとったとき、そうなるようである．

問題 D1～D3 : al=1.2 (nc=30), eps=$10^{-9}$, m=160



最後に、k値が一様な場合にこのプログラムを使用したらどうなるかを見ておく．下のものは、問題Cの場によるものから作成したが、k値が一様となってしまえば、どれでも同じようなものである．epsbiの最適値が $(al/nc)*5$ のあたりに移動することがこれで判る． $al=1.5$ , $nc=25$ である．



6　初期ベクトルについて.

今回のテスト例では、初期ベクトルとしては、

1)　各グループの先頭ベクトルにはウイルキンソンの初期ベクトル，即ち、　$A-\alpha I = LU$　と分解したとき、

$$v = U^{-1}e \quad , \text{但し} \ e = (1, 1, \cdots, 1)^T .$$

2)　その他のものとしては、各 $k$（$= 1, \cdots, nc$ の何れか）に対し、両端開放の一次元モデル（$-u'' = \lambda u$）を $n$ 等分した差分解による固有ベクトル $v_k$（$k = 1, \cdots, nc$）を与える.

とした.

実際の応用においては、もっと好適な初期ベクトルを与えることが、しばしば可能であろう. 例えば、今の問題にそくして言うならば、構造の同じ熱伝導場で、$k$ 値の分布をいろいろ変えたとき、どうなるかの一連の熱伝導問題を解きたいとする. 例えば、問題 D3 ～D1 の問題を次々に解きたい，とする. このとき、問題 Dm の解を問題 Dm-1 の初期ベクトルに使うというやり方が当然考えられる.

勿論、全く言葉通りにやったのでは、うまく行くときもあるがうまく行かないことがある. 何故ならば、グループ分けまでのところは各問題毎に独立にやるわけゆえ、たとえば問題D2の第2グループの一本が、問題D1 の第1グループに編入されることがある. D2の第し番目の固有値に対する固有

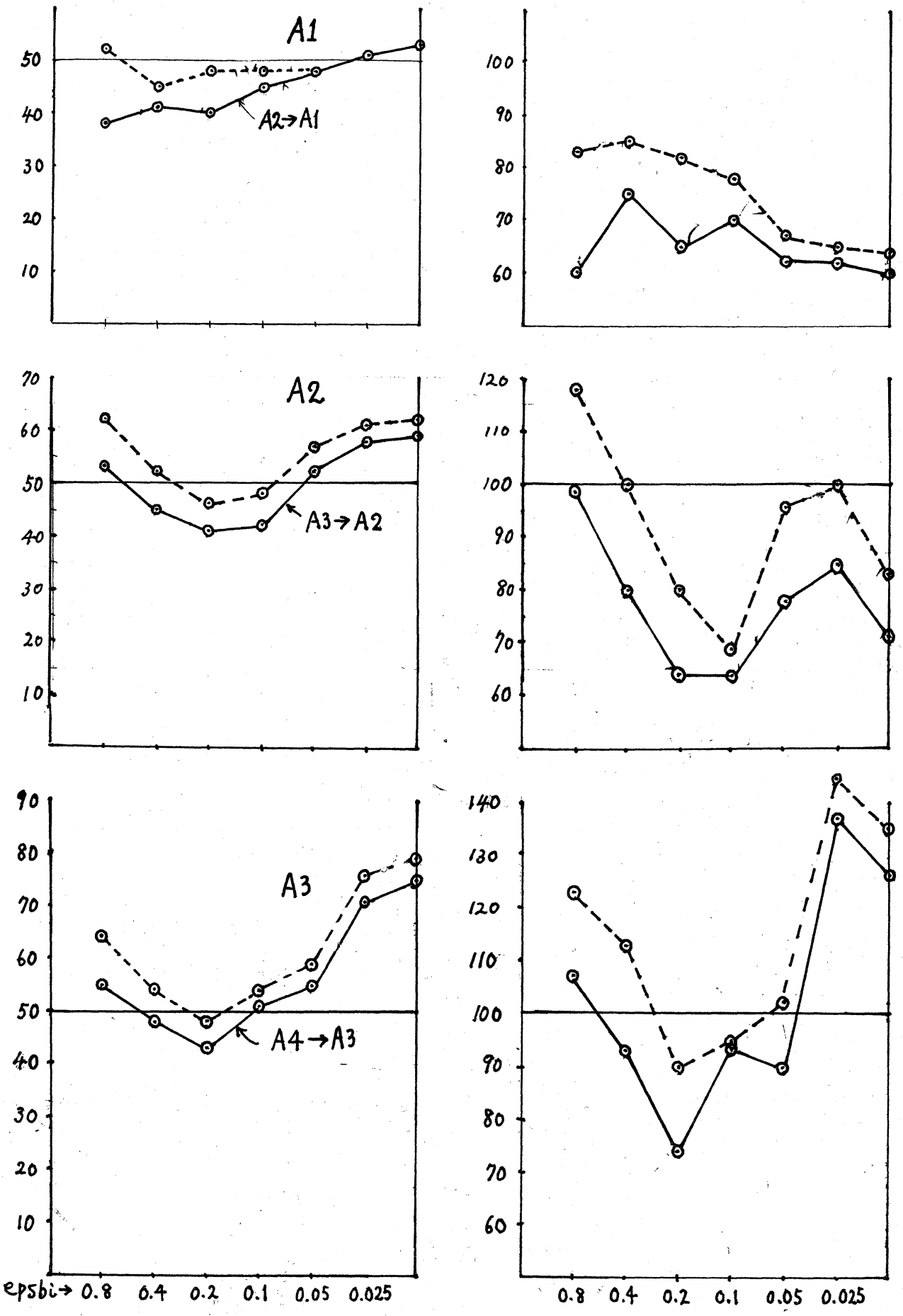
ベクトルがD1の半径-1番目とか径-2番目の固有値に対応する固有ベクトルに移る，ということはしばしばあることである．こういうことがあっても、それなりに動作するよう手当てをしておかねば、こういう使い方を汎用的に奨めるわけには行かない．

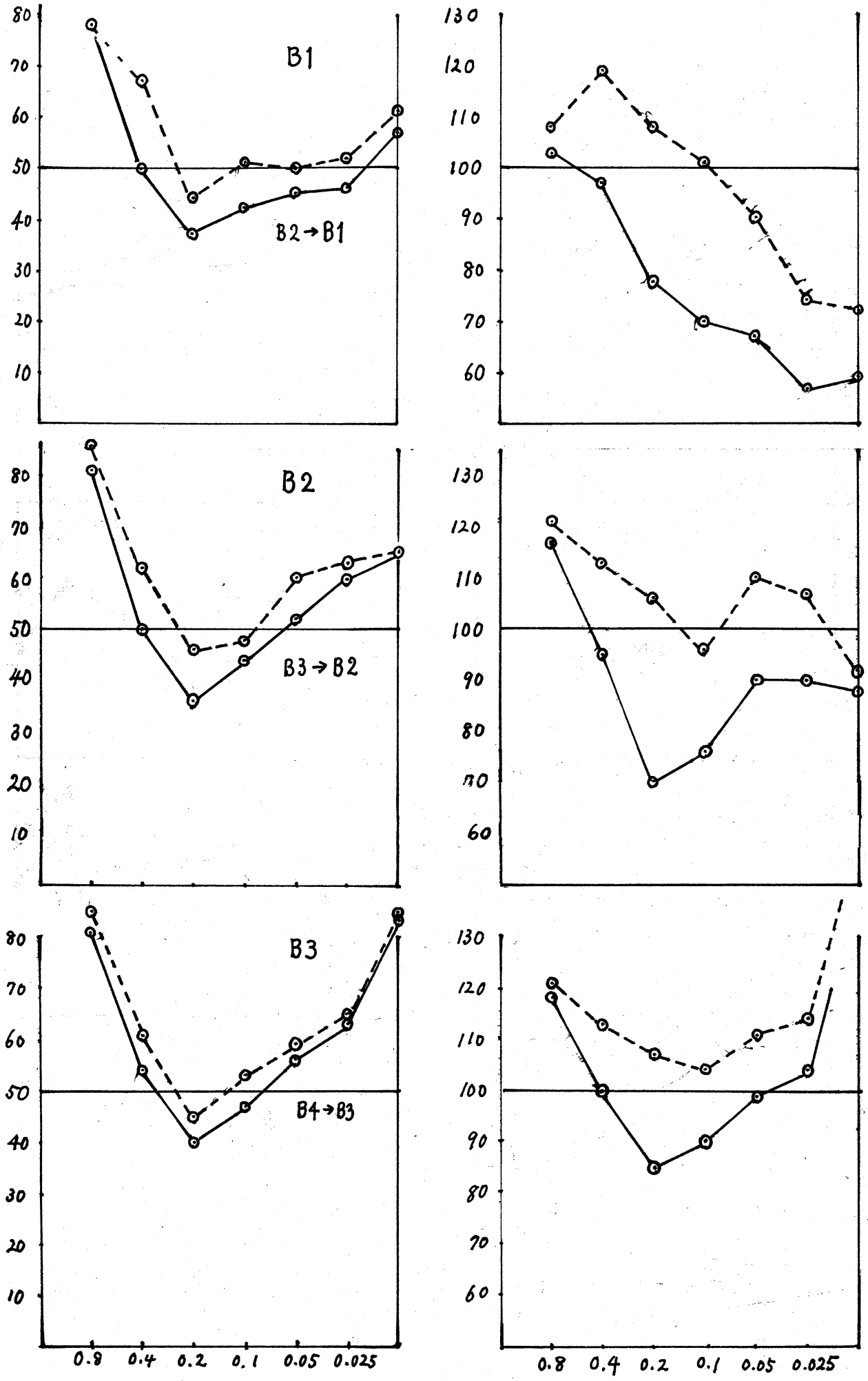
ひとつの手当てとして我々はとりあえず次のようにした：

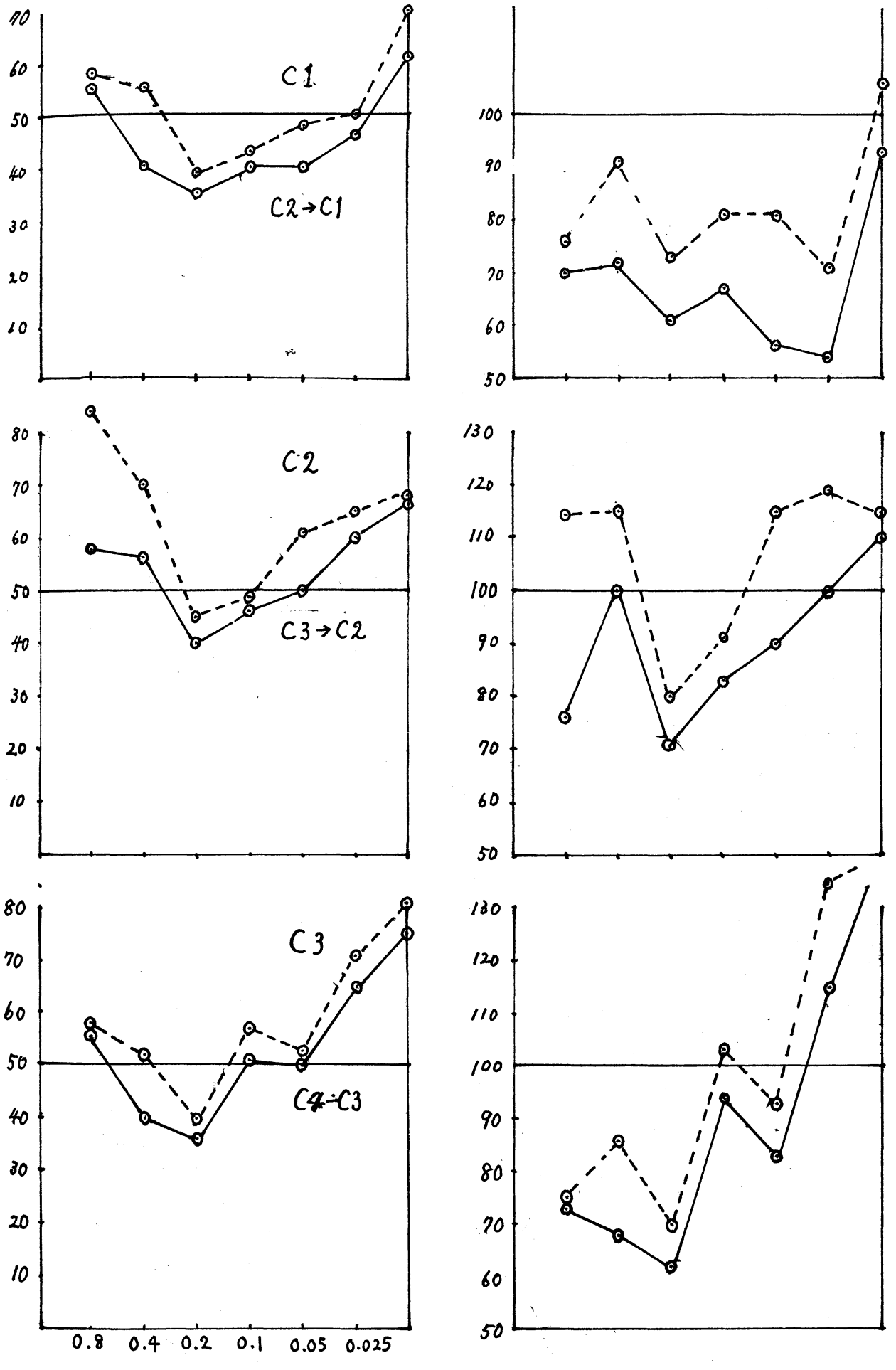
1) $ln[ig]=1$ 即ち単純固有値に対してはウイルキンソンの初期ベクトルを与える．即ち $v=U^{-1}e$ ．

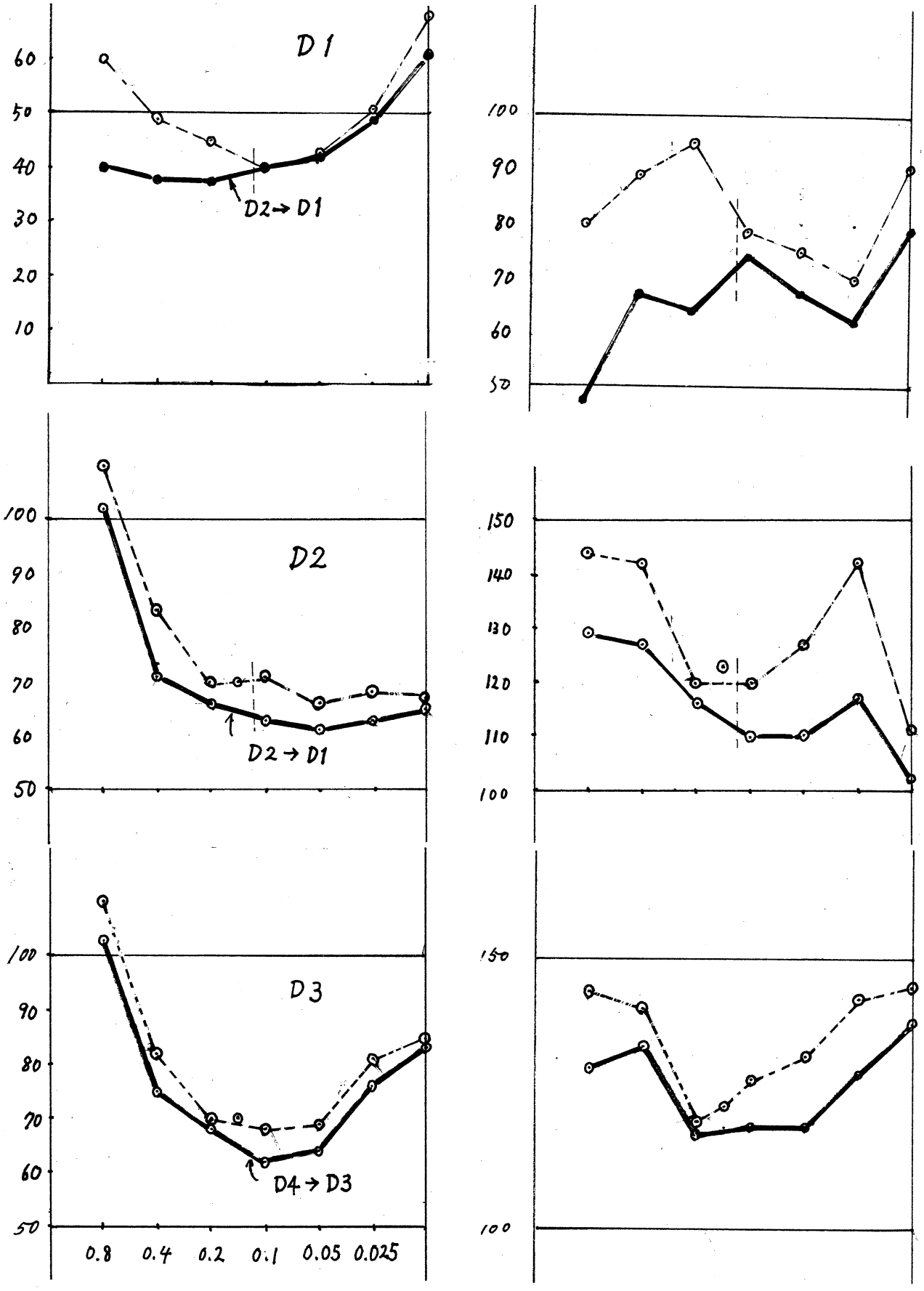
2) $ln[ig]>1$ のグループに対しては、$v_j$ として、対応する番号の固有ベクトル $w_j$ を機械的に与えるが、ひとつ前のグループの既求の固有ベクトルと直交化を行なう過程で、もしそれらと‘殆んど平行’なものであることが判ったら、それを‘乱数ベクトル’ととり換える．

このような方法で、どの程度うまく行くかを調べたのが、次頁のデータである．点線が単独試作のときのもの，実線が、今の応用動作を行なったときのものである。違うのは同時逆反復回数だけだから、$n$が小さい問題の方が利きが著しい筈であるので $n=80$ として評価した．それにしても意外に利益はすくない．


A1
A2→A1
A2
A3→A2
A3
A4→A3
epsbi→ 0.8 0.4 0.2 0.1 0.05 0.025
0.8 0.4 0.2 0.1 0.05 0.025


B1
B2→B1
B2
B3→B2
B3
B4→B3
80
70
60
50
40
30
20
10
130
120
110
100
90
80
70
60
0.8
0.4
0.2
0.1
0.05
0.025


C1
C2→C1
C2
C3→C2
C3
C4→C3
0.8
0.4
0.2
0.1
0.05
0.025


D1
D2→D1
D2
D2→D1
D3
D4→D3
0.8
0.4
0.2
0.1
0.05
0.025

## 7 おわりに、

はじめにのべたように、今回開発したスツルム・同時逆反復法は、従来からあるスツルム・逆反復法系統のものと同時逆反復法系統のものそれぞれがもつ長所を生かし、欠点を表面化させないように両者を統合して一つのプログラムにまとめ上げたものである。今回のテストは $Ax=\lambda x$ 形の問題ながら、行列Aとしては、問題になりそうな可なり広い範囲にわたるケースをえらんで、プログラムのタフさを確かめた。$Ax=\lambda Mx$ 形の問題に対しては、Mが正定値であっても特異に近いとき、それなりの配慮を追加せねばなるまい。

場の問題 $div(-k(x)\nabla u)=\lambda u$ の'拡散係数' $k(x)$ が問題毎にちがうような一連の問題を解くとき、このプログラムをうまく利用できるかどうかについてはすこしばかりふれたが、この点に関しては、実用上重要ゆえ、さらに検討を重ねる価値がある。これとよく似た応用として、場の問題を一度に解くのでなく、予じめ粗い有限要素分割で解いたものを、精しい分割によるものの初期値作成に利用する，という考え方がある。これは誰もが考えることながら、手法を汎用化するためには、理論的にも深い検討を重ねながらの実務的な手当を要するであろう。これも今後の課題としたい。

# 文献

(1) 村田 '対称帯行列固有値解析' 数理解析研講究録 422号

(2) Bathe & Ramawamy 'An accelerated subspace iteration Method' Comp. Meth. Appl. Mech. Eng. 23 (1980) 313~331

(3) [Wilson 1978] in Parlett (6)

(4) Jennings & Agar 'Hybrid Sturm Sequence and Simultanious Iteration Methods' in Proc. Symp. on Applic. of Comp. meth. in Eng. (1978). Univ. Southern California Press.

(5) Wilkinson-Reinsch, (1971) 'Handbook' (Springer) にのっている. Contribution I/6 もとの論文は Numer. Math. 9, 279~301 (1967)

(6) Parlett 'The Symmetric Eigenvalue Problems' (1980) Prentice H.